BACKGROUND

Questions:

Q: What would a buyer not expect when receiving a new car from the showroom?

A: A scratch down the side of the vehicle.

Q: What essential components are expected in a new car?

A: Five wheels, a steering wheel, an engine, and all other essential components.

Q: What should accompany the delivery of a new car?

A: Appropriate documentation and completed, satisfactory pre-sales checks.

Q: How should the delivered car match the sales literature?

A: It should have the correct engine size, colour scheme, extras, and performance figures.

Q: What creates the buyer's level of expectation for a new car?

A: Brochures, experience of sitting in the driving seat, and a test drive.

Q: How does a buyer feel if car expectations are not met?

A: Justifiably aggrieved.

Q: What is a common issue with new software installations?

A: They are often delivered not working as expected or not working at all.

Q: What is identified as a major contributing factor to faulty software delivery?

A: The inadequacy of the testing to which software applications are exposed.

Q: How is software testing described in terms of complexity and implementation?

A: It is neither complex nor difficult to implement.

Q: What is lacking in most software testing practices?

A: The necessary rigour to provide confidence in delivered software.

Q: Why is software testing considered costly?

A: It requires human effort or technology to multiply the effect of human effort.

Q: At what level is software testing usually implemented?

A: At a level that does not provide assurance that software will operate effectively, efficiently, or correctly.

Q: What does the book aim to provide regarding software testing?

A: A basis for constructing a practical and cost-effective software testing regime.

INTRODUCTION

Questions:

Q: What is the first objective of the opening chapter?

A: To introduce fundamental ideas that underpin software testing and explain new terminology.

Q: What is the second objective of the opening chapter?

A: To establish the structure used throughout the book for learning and revision.

Q: What is the third objective of the opening chapter?

A: To point forward to the content of later chapters.

Q: What does the chapter begin by defining?

A: It begins by defining what readers are expected to gain from reading the chapter.

Q: What are the learning objectives in the chapter based on?

A: They are based on the Software Foundation Certificate syllabus (ISTQB, 2010).

Q: What must readers ensure before attempting the examination?

A: They must ensure they have achieved all the learning objectives.

WHY SOFTWARE FAILS

Questions:

Q: What caused the failure of the Ariane 5 rocket in June 1996?

A: A software error caused the rocket to deviate from its vertical ascent, triggering self-destruction.

Q: What issue occurred with the UK Government's online tax return system?

A: A user could sometimes see the income of a previous user.

Q: Why did the UK's top 10 wanted criminals website go offline in 2005?

A: The website couldn't handle the high traffic generated by media publicity.

Q: What problem occurred when a book retailer allowed negative book orders?

A: Negative orders led to refunds being issued to the 'purchaser'.

Q: What was the result of a small change in the billing system of an electrical provider?

A: It caused a blackout in a major US city.

Q: What makes the listed software failure examples startling?

A: They suggest that obvious issues were missed and systems launched before being ready.

Q: Why are software systems sometimes not adequately tested?

A: Because of pressures like deadlines, system complexity, and changing technology.

Q: Where do major system failures usually begin?

A: They begin with errors in specifications, designs, or software code.

Q: What happens if a faulty component is built into a system?

A: The system may fail.

Q: How can environmental conditions cause system failure?

A: Conditions like radiation, magnetism, electronic fields, or pollution can affect hardware and firmware.

Q: How can we avoid system failures?

A: By avoiding or finding and rectifying errors and faults.

Q: When should testing begin in the development process?

A: Testing should begin right at the start when errors are first made.

Q: Until when should testing continue?

A: Until we are confident there will be no serious system failures.

Q: What are some potential consequences of incorrect software for people?

A: It can cause deaths, such as in aircraft crashes or life support failures.

Q: How can incorrect software affect companies?

A: It can lead to incorrect billing and financial losses.

Q: How can software errors harm the environment?

A: They can release chemicals or radiation into the atmosphere.

Q: What is an example of a software failure affecting all three areas: people, companies, and the environment?

A: A train carrying nuclear waste crashing due to system failure.

Q: What are the possible outcomes of software failure?

A: Loss of money, loss of time, loss of business reputation, injury, and death.

KEEPING SOFTWARE UNDER CONTROL

(Exhaustive testing of complex systems is not possible)

Questions:

Q: What common theme is identified in the software failure examples?

A: Either insufficient testing or the wrong type of testing was done.

Q: Why is achieving more and better software testing not simple?

A: Because exhaustive testing of complex systems is not possible.

Q: What was reused in the Ariane 5 rocket launch?

A: A software module from the Ariane 4 programme.

Q: What issue arose from reusing the Ariane 4 module in Ariane 5?

A: Unused functionality caused a directional nozzle to move uncontrollably due to incorrectly updated variables.

Q: Why did the Ariane 5 failure occur even though the faulty code wasn't used?

A: Unexpected conditions triggered a malfunction in unused software.

Q: What principle of software testing is illustrated by the Ariane 5 example?

A: It is impossible to test everything exhaustively in large and complex systems.

Q: What would have happened if every possible test had been run on Ariane 5?

A: The problem would have been detected, but the launch would never have occurred.

Q: Why is it unhelpful to say not enough testing was done in Ariane 5?

A: Because the right kind of testing was not done to detect the specific problem.

Testing and risk

Questions:

Q: What is inherent in all software development?

A: Risk is inherent in all software development.

Q: What are examples of uncertainties in software development?

A: The system may not work or the project may not be completed on time.

Q: When do uncertainties become more significant?

A: As system complexity and the implications of failure increase.

Q: Why would an automatic flight control system be tested more than a video game?

A: Because the risk is greater.

Q: What does greater system complexity imply?

A: A greater probability and impact of failure.

Q: What should the extent and focus of testing be related to?

A: It should be related to the level of risk.

Q: What does greater risk imply in terms of testing?

A: More and better testing.

Testing and quality

Questions:

Q: What is a good starting point for defining quality in a system?

A: If a system meets its users' requirements.

Q: What was the functional weakness in the online tax returns system?

A: It allowed one user to view another user's details.

Q: What kind of failure occurred in the top 10 criminals website example?

A: A non-functional failure.

Q: Why did the top 10 criminals website fail to deliver its services?

A: It was not designed to handle the peak load after media coverage.

Q: What must the software development process balance?

A: Competing demands for time, money, and quality.

Q: What does the triangle of resources consist of?

A: Time, money, and quality.

Q: What do time, money, and quality influence in software development?

A: They influence the features included in the delivered software.

Q: What is one role of testing before a system enters service?

A: To examine key functional and non-functional requirements and report defects.

Q: Can testing directly remove defects or enhance quality?

A: No, testing cannot directly remove defects or enhance quality.

Q: How does testing contribute to improved quality?

A: By reporting defects for removal and enabling quality measurement.

Q: What does systematic testing allow?

A: Measurement of some aspects of software quality.

Q: What is testing a component of in the development process?

A: It is a component of overall quality assurance activity.

Deciding when ‘enough is enough’

Questions:

Q: How do we decide how much testing is enough?

A: By considering risk, acceptable quality levels, and prioritisation.

Q: Why can't we test everything in a system?

A: Because it is not feasible due to limitations in time and resources.

Q: What is the most important aspect of achieving acceptable results from limited testing?

A: Prioritisation.

Q: Why is prioritisation important in testing?

A: It ensures the most important tests are done first.

Q: What should the most important tests focus on?

A: The most important functions, non-functional behaviour, and significant risks.

Q: What helps decide when it is safe to stop testing?

A: Setting objective completion criteria.

Q: What do completion criteria define?

A: How much software is to be tested and what defect levels are acceptable.

Q: What do priorities and completion criteria help with?

A: They provide a basis for planning the testing process.

Q: What may need to be compromised in the end?

A: The desired level of quality and risk.

Q: How does prioritisation protect the testing process from time or effort reductions?

A: It ensures the most important tests are completed first.

WHAT TESTING IS AND WHAT TESTING DOES

(Testing and debugging)

Questions:

Q: What is the main purpose of software testing?

A: To reduce risk and improve quality by finding defects.

Q: How is debugging defined in the passage?

A: Debugging is the process developers use to identify and correct bugs or defects in code.

Q: What is the primary focus of testing?

A: Systematic exploration to find and report defects.

Q: Does testing include fixing defects?

A: No, defects are passed to the developer to correct.

Q: What does testing ensure after a correction is made?

A: Those changes are checked for their effect on other parts of the system.

Q: Why is effective debugging important before testing?

A: To raise the quality level to a point where rigorous testing is worthwhile.

Q: Does debugging confirm that all requirements are met?

A: No, debugging does not give confidence that requirements are completely met.

Q: What does testing provide that debugging does not?

A: A rigorous examination of system behaviour and defect reporting.

Q: What happens after defects are corrected?

A: Testing repeats enough tests to ensure the corrections are effective.

Q: Why are both testing and debugging necessary?

A: Because both are needed to achieve a quality result.

Static testing and dynamic testing

Questions:

Q: What is static testing?

A: Static testing is testing where the code is not exercised.

Q: Where do failures often begin according to the passage?

A: Failures often begin with a human error in a document such as a specification.

Q: Why should testing start as early as possible?

A: Because errors are much cheaper to fix than defects or failures.

Q: What technique is mentioned as part of static testing?

A: Reviews are a technique used in static testing.

Q: What can reviews help prevent?

A: Reviews can help prevent defects by removing ambiguities and errors.

Q: What is dynamic testing?

A: Dynamic testing exercises the program under test with some test data.

Q: What do we refer to in the context of dynamic testing?

A: We refer to test execution in the context of dynamic testing.

Q: What types of testing does software testing include?

A: Software testing includes both static and dynamic testing.

Testing as a process

Questions:

Q: What must be done before test execution?

A: Preparatory work to design the tests and set them up must be done before test execution.

Q: What must be done after test execution?

A: Results must be recorded and checked to see if the tests are complete.

Q: What is more important than the execution process in testing?

A: Deciding what we are trying to achieve and setting clear objectives is more important.

Q: How would a test for confidence in specifications differ from one aiming to find defects?

A: It would be designed differently based on its objective.

Q: Why do we define a test process?

A: To ensure we do not miss critical steps and do things in the right order.

Testing as a set of techniques

Questions:

Q: What is considered a good test in software testing?

A: A good test is one that finds a defect if there is one present.

Q: What value does a test that finds no defect add?

A: It adds no value but consumes resources.

Q: What opportunity does a test that finds a defect provide?

A: It creates an opportunity to improve the quality of the product.

Q: What is the first method used to design effective tests?

A: The first method is using well-proven test design techniques.

Q: Where are the most important test design techniques explained?

A: They are explained in detail in Chapter 4.

Q: What is the second mechanism used to ensure test effectiveness?

A: The second mechanism is applying testing principles.

Q: What can guide testing when rigorous test design isn't possible?

A: General testing principles can still guide the testing.

GENERAL TESTING PRINCIPLES

(Testing shows the presence of bugs)

Questions:

Q: What is the main purpose of testing according to the passage?

A: The main purpose of testing is to find defects.

Q: Can testing prove that software is error free?

A: No, testing cannot show that the software is error free.

Q: What does running a test through software demonstrate?

A: It can only show that one or more defects exist.

Q: Was the failure of the top 10 wanted criminals website due to functional defects?

A: No, the failure was non-functional despite the absence of functional defects.

Q: What does the top 10 criminals website example demonstrate about defect absence?

A: It shows that absence of defects is not enough to release a system into operation.

Q: What can retesting demonstrate under the same conditions?

A: Retesting can show that a specific reported problem no longer exists.

Q: What should tests be designed to achieve?

A: Tests should be designed to find as many defects as possible.

Exhaustive testing is impossible

Questions:

Q: Why is exhaustive testing not possible for large complex systems?

A: Because there are too many possible input combinations and circumstances.

Q: Can small pieces of software be tested exhaustively?

A: It is very difficult due to the high number of possible inputs and scenarios.

Q: Why is the number of input permutations higher than 26 × 26 × 26?

A: Because standard keyboards allow 256 × 256 × 256 permutations.

Q: What complicates testing even simple password input?

A: Factors like special key combinations, delays between inputs, and timing variations.

Q: What does entering characters with a long delay demonstrate?

A: That the same inputs under different conditions can produce different results.

Q: Can we determine if defects exist without trying all combinations?

A: No, we can't say whether defects exist without testing all combinations.

Q: Why are risk and priorities important in testing?

A: They help focus on the most important aspects to test when exhaustive testing is not possible.

Early testing

Questions:

Q: Why is early testing important?

A: It helps prevent time pressure from squeezing testing and allows for more time to find defects.

Q: When can testing begin in the software development life cycle (SDLC)?

A: Testing can begin as soon as work-products, such as requirement documents, are available.

Q: What role do requirement documents play in early testing?

A: They serve as the basis for acceptance testing, allowing the creation of acceptance tests early on.

Q: What can early testing highlight about requirements?

A: It can reveal whether individual requirements are testable or if they are ambiguous or missing.

Q: What is a common cause of software system problems?

A: Many problems are traced back to missing or incorrect requirements.

Q: What is the purpose of reviews in early testing?

A: Reviews help identify and correct errors before they become defects in later stages.

Q: What does Ed Kit's work discuss regarding errors in the SDLC?

A: It emphasizes identifying and eliminating errors in the stage where they are introduced to avoid "errors of migration."

Q: What happens if a defect is not corrected at its origin in the SDLC?

A: It leads to rework in multiple stages, resulting in the "cost escalation model."

Q: What happens to the cost of fixing defects the later they are identified?

A: The cost of fixing defects rises steeply as they are identified later in the development process.

Q: What is the main point about testing objectives in different stages?

A: Each stage of testing has defined objectives, which may vary depending on the focus of that stage.

Defect clustering

Questions:

Q: What is defect clustering in software testing?

A: It refers to the phenomenon where most defects are concentrated in a small number of modules in a large application.

Q: What causes defect clustering in software?

A: It can be caused by system complexity, volatile code, the effects of change, and the experience level of development staff.

Q: What is the Pareto principle in software testing?

A: The Pareto principle suggests that approximately 80% of the problems are found in about 20% of the modules.

Q: How can testing activity benefit from defect clustering?

A: Testing should target areas where a high proportion of defects are found, focusing on the modules with the most issues.

Q: Should testing only focus on the modules with the most defects?

A: No, testing should not exclusively concentrate on these areas; the remaining code should also be searched for defects.

The pesticide paradox

Questions:

Q: What is the pesticide paradox in software testing?

A: The pesticide paradox is the idea that running the same set of tests continually will eventually stop finding new defects.

Q: Why does the pesticide paradox occur in software testing?

A: It occurs because developers anticipate the test conditions and will test them before software is delivered, reducing the effectiveness of repeated tests.

Q: What happens if the same test set is used repeatedly in software testing?

A: The tests become less effective over time, as they no longer find new defects in the software.

Q: What is regression testing?

A: Regression testing is the practice of testing a small change in software to ensure that no new defects have been introduced.

Q: Why might regression tests fail to detect issues in production?

A: Regression tests may no longer be relevant to the updated system requirements or objectives.

Testing is context dependent

(Absence of errors fallacy)

Questions:

Q: Why is testing considered context dependent?

A: Because different types of applications require different approaches to testing based on their purpose and usage.

Q: How does testing differ between a basic website and an e-commerce site?

A: A basic website may require simple viewing tests, while an e-commerce site needs more rigorous tests, especially for security.

Q: Why does an air traffic control system require more rigorous testing than a mortgage calculator?

A: Because the potential risk and consequences of failure are much higher in an air traffic control system.

Q: What role does risk play in determining the type of testing needed?

A: Higher risk increases the need to invest more in thorough testing before implementation.

Q: Why is security testing especially important for an e-commerce site?

A: To ensure users cannot bypass passwords or use invalid credit card data.

Q: Why might some types of testing require specialist staff and tools?

A: Because certain areas like security testing are complex and not suitable for all applications..

FUNDAMENTAL TEST PROCESS

Questions:

Q: What is the fundamental test process?

A: It is a key element of testing that applies at all stages and includes five main activities.

Q: What is the most visible part of testing?

A: Test execution.

Q: Why are planning and analysing important in testing?

A: Because they enhance and amplify the benefits of test execution.

Q: What are the five parts of the fundamental test process?

A: Planning and control, analysis and design, implementation and execution, evaluating exit criteria and reporting, test closure activities.

Q: Are the fundamental test process activities done in a rigid sequence?

A: No, they are not undertaken in a rigid way.

Q: What might happen if a defect is found during test execution?

A: It might require adding missing functionality, which then needs to be tested.

Q: Why might earlier activities need to be revisited during testing?

A: Because new features or corrections may need additional analysis and design.

Q: Can test execution begin before all tests are designed?

A: Yes, due to time pressure, execution can begin before all tests are fully designed.

Test planning and control

Questions:

Q: What does test planning determine?

A: It determines what is going to be tested and how it will be achieved.

Q: What is defined during test planning?

A: The test completion criteria.

Q: What is the purpose of test completion criteria?

A: To determine when testing is finished.

Q: What does test control involve?

A: Comparing progress against the plan and adjusting as needed.

Q: When do we undertake planning and control?

A: Throughout the testing activities.

Q: What is monitoring in the context of test control?

A: Measuring what has happened during testing.

Q: What is control in the context of test activities?

A: Adjusting future activities based on experience.

Q: How do monitoring and control relate to planning?

A: They feed back into the continual activity of planning.

Test analysis and design

Questions:

Q: What are analysis and design concerned with in the testing process?

A: They are concerned with what to test (test conditions) and how to combine them into test cases.

Q: What is the purpose of combining test conditions into test cases?

A: To cover as many test conditions as possible with a small number of test cases.

Q: What stage serves as the bridge between planning and test execution?

A: The analysis and design stage.

Q: What does the analysis and design stage look backward and forward to?

A: It looks backward to planning and forward to execution activity.

Q: What must be considered during the test design process?

A: The test data required for the test conditions and test cases.

Q: Why is it important to detail expected outcomes before test execution?

A: To avoid missing a vital detail that is wrong.

Q: What is one example of a trivial expected outcome during testing?

A: Money should not be refunded to a customer's card without supervisor intervention.

Q: What does reviewing the test basis include?

A: Reviewing requirements, architecture, design, interface specifications, and other parts.

Q: What does analysing test items help identify?

A: Test conditions and test data required.

Q: What does designing the tests include?

A: Assigning priority.

Q: What must be determined regarding requirements and systems during test design?

A: Whether they are testable.

Q: What should be detailed regarding the test environment?

A: What it should look like and whether infrastructure and tools are required.

Q: What must be highlighted for test conditions and test cases?

A: The test data required.

Q: What should be created between test basis and test cases?

A: Bi-directional traceability.

Test implementation and execution

Questions:

Q: What does test implementation and execution involve?

A: It involves running tests and performing necessary set-up or tear-down activities.

Q: What must be checked before testing begins?

A: The test environment must be checked.

Q: What is the most visible part of testing?

A: Test execution.

Q: How are the most important tests determined?

A: They are determined during planning and refined during test design.

Q: Why are test cases combined into an overall run procedure?

A: To utilise test time efficiently.

Q: Why is the logical ordering of tests important?

A: So the outcome of one test can create preconditions for later tests.

Q: What needs to be done as tests are run?

A: Log the outcome and compare expected results with actual results.

Q: What should happen if there is a discrepancy between expected and actual results?

A: It should be investigated and a test incident raised if necessary.

Q: What are the two types of testing required when changes are made?

A: Retesting and regression testing.

Q: What does retesting focus on?

A: The changed area of functionality in fine detail.

Q: What does regression testing aim to ensure?

A: That no unintended changes have occurred elsewhere.

Q: What are examples of processes to include in a regression test pack for a financial system?

A: End of day, end of month, and end of year processing.

Q: What are key parts of test implementation and execution?

A: Developing test cases, creating test data, writing procedures, preparing test harnesses, and automating scripts.

Q: Why are test cases collected into test suites?

A: So they can be run one after another efficiently.

Evaluating exit criteria and reporting

Questions:

Q: What are exit criteria?

A: They are conditions defined during test planning to determine when testing is complete.

Q: When are exit criteria checked?

A: They are checked at the end of test execution.

Q: Who checks if the exit criteria have been met?

A: The test manager.

Q: What happens if the exit criteria of 85 per cent statement coverage results in only 75 per cent?

A: Either the exit criteria are changed or more tests are run.

Q: Can more tests be required even if exit criteria are met?

A: Yes, more tests may still be required.

Q: What should a test summary for stakeholders include?

A: What was planned, what was achieved, differences, and untested areas.

Q: What is the purpose of evaluating exit criteria?

A: To check if the criteria have been met, decide on further testing, and report results.

Q: Who are the test results written for?

A: The business sponsors and other stakeholders.

Test closure activities

Questions:

Q: What is the focus of test closure activities?

A: Ensuring everything is tidied away, reports are written, and defects are closed or deferred.

Q: What happens to deferred defects during test closure?

A: They are clearly marked for another phase.

Q: What does ensuring documentation is in order involve?

A: Defining what was delivered, closing incidents, raising changes, and documenting acceptance.

Q: What is done with the test environment during test closure?

A: It is closed down and archived.

Q: What is passed to the maintenance team during test closure?

A: Testware is passed to the maintenance team.

Q: What is documented for future improvement in test closure?

A: Lessons learned are written down to improve the testing process.

Q: What is the purpose of writing down lessons learned?

A: To improve the testing process and increase testing maturity.

THE PSYCHOLOGY OF TESTING

Questions:

Q: Who may be involved in the total testing effort?

A: Developers, professional testers, specialists, and users assisting with acceptance testing.

Q: Why is it better if testing is not done by the person who wrote the code?

A: Because the creator may not see flaws in their own work.

Q: What mindset change is needed for developers to test their own code?

A: They must switch from proving it works to trying to show it does not work.

Q: What is test independence?

A: Having software tested by someone not involved in its creation.

Q: Who provides the least independent testing?

A: Those who wrote the code.

Q: Who provides the most independent testing?

A: Members of a different company like a testing consultancy.

Q: What is a disadvantage of using highly independent testers?

A: It is more expensive.

Q: Why might developers testing their own code be useful?

A: Problems can be fixed immediately without extensive error logs.

Q: What is a difficulty in developers testing their own code?

A: It is hard to find their own mistakes.

Q: What influences how people test software?

A: Their objectives and project goals.

Q: How might a software author view a defect report?

A: As a personal criticism.

Q: How should defect reports be presented?

A: With tact and diplomacy, focusing on the software, not the individual.

Q: Where might a mistake originate besides the code?

A: In requirement documents or system specification.

Q: What is the aim of raising defects constructively?

A: To avoid bad feelings.

Q: How should testers and developers work together?

A: As a team with the joint goal of better quality systems.

Q: How should communication be expressed?

A: Objectively and in impersonal ways.

Q: What is a key part of good communication?

A: Understanding and being understood.

Q: What is a shared goal of testers and developers?

A: Better quality systems delivered in a timely manner.

Q: What supports the goal of delivering quality systems?

A: Good communication between testers and developers.

CODE OF ETHICS

Questions:

Q: What are testers required to adhere to before moving on to detailed topics?

A: Testers must adhere to a code of ethics.

Q: Why must testers act in a professional manner?

A: Because they can have access to confidential and/or privileged information.

Q: Who must treat information with care and act responsibly to its owners?

A: Testers.

Q: Who does the code of ethics apply to?

A: Those who have achieved software testing certification.

Q: What should certified software testers consider in their actions?

A: The wider public interest.

Q: How should certified software testers act regarding their client and employer?

A: In the best interests of their client and employer.

Q: What standard should certified software testers meet in their deliverables?

A: The highest professional standards possible.

Q: How should certified software testers maintain their judgement?

A: With integrity and independence.

Q: What should certified test managers and leaders promote in management?

A: An ethical approach to the management of software testing.

Q: How should certified software testers treat their profession?

A: They shall advance the integrity and reputation of the profession.

Q: What is expected of certified testers towards their colleagues?

A: They shall be fair to and supportive of their colleagues.

Q: What should certified testers promote with software developers?

A: Cooperation.

Q: What is expected from certified testers regarding their professional growth?

A: They shall participate in lifelong learning.

Q: How should certified testers approach their profession?

A: They shall promote an ethical approach to the practice of the profession.

Q: What does the code of ethics interact with according to the passage?

A: Other areas of the syllabus.

Q: What is the reason for the whole book according to the passage?

A: The implementation of the code of ethics.